

Laboratorio No. 3

Enlace al repositorio: <https://github.com/bl33h/lstm>

✓ [7 marks]

✓ [7 marks]

✓ [7 marks]

✓ [7 marks]

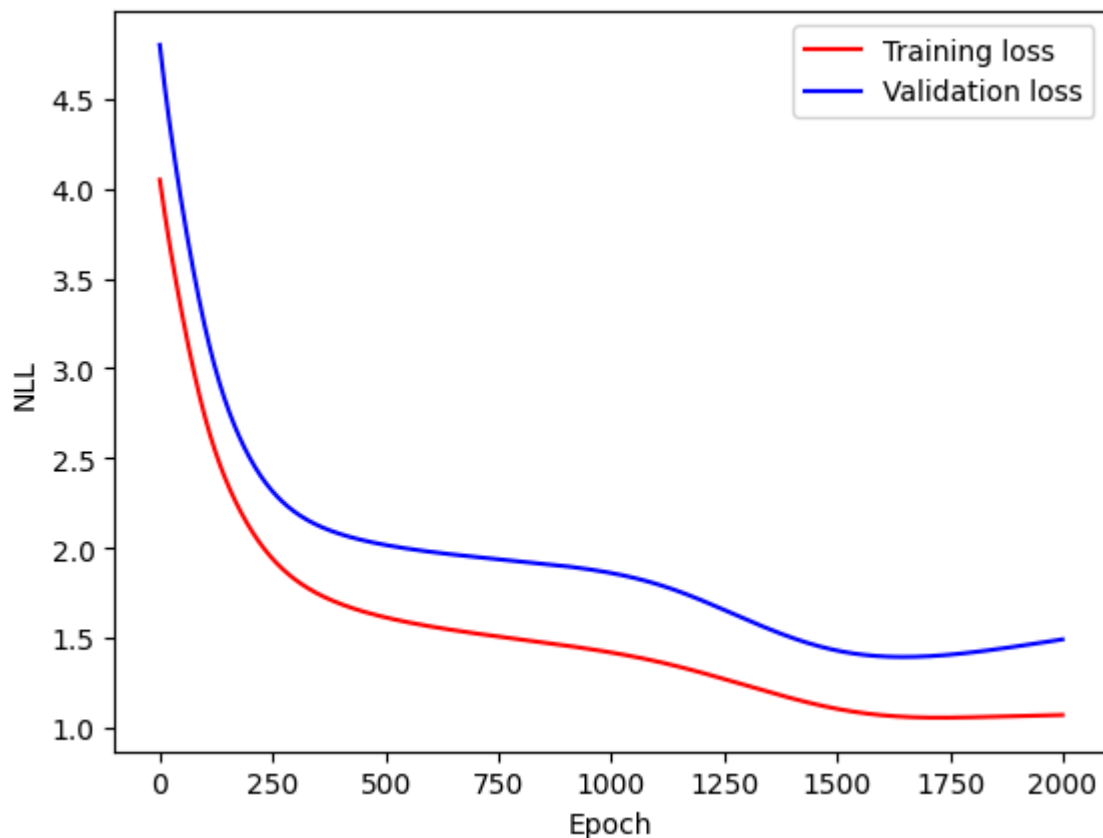
```
print()  
print("La fraccion de abajo muestra su rendimiento basado en las partes visibles de este laboratorio")  
tick.summarise_marks() #
```

✓ 0.0s

La fraccion de abajo muestra su rendimiento basado en las partes visibles de este laboratorio

158 / 158 marks (100.0%)

Parte I



- ¿Qué interpretación le da a la separación de las graficas de training y validation?
 - Los gráficos manifiestan una separación entre las curvas de pérdida de entrenamiento y validación a lo largo de las épocas, lo que puede sugerir la existencia de overfitting.
- ¿Cree que es un buen modelo basado solamente en el loss?
 - A criterio propio, considero que este modelo no parece ideal debido a que no generaliza bien a datos nuevos. Sin embargo, se podrían emplear estrategias para combatirlo.
- ¿Cómo deberían de verse esas gráficas en un modelo ideal?
 - En términos generales se deberían evidenciar las pérdidas de entrenamiento y validación de una manera similar, converger a valores bajos y mantenerse cercanas sin separación.

Parte II

- ¿Qué modelo funcionó mejor? ¿RNN tradicional o el basado en LSTM? ¿Por qué?
 - El RNN clásico fue más efectivo, a pesar de su entrenamiento más lento. Detecta mejor los cambios entre las letras, mientras que el LSTM a menudo coloca "EOS".
- Observen la gráfica obtenida arriba, ¿en qué es diferente a la obtenida a RNN? ¿Es esto mejor o peor? ¿Por qué?
 - Los gráficos manifiestan que el RNN tradicional muestra una pérdida más estable, mientras que el LSTM tiene una pendiente más pronunciada y más oscilaciones, lo que puede indicar inestabilidad.
- ¿Por qué LSTM puede funcionar mejor con secuencias largas?
 - Se debe al mejor manejo de la memoria y el olvido, lo que los hace adecuados para secuencias extensas.

Parte III

- Compare las graficas obtenidas en el LSTM "a mano" y el LSTM "usando PyTorch, ¿cuál cree que es mejor? ¿Por qué?
 - El gráfico de PyTorch es superior. Muestra una pérdida más pronunciada y menos oscilaciones, indicando un mejor ajuste y disminución de overfitting. Estos dos aspectos son clave pues impactan en el resultado obtenido en cada caso, lo que también puede ser considerado como una confirmación de lo anteriormente mencionado.
- Compare la secuencia target y la predicha de esta parte, ¿en qué parte falló el modelo?
 - El modelo casi acierta, pero se equivocó en la transición entre caracteres, apresurándose.
- ¿Qué sucede en el código donde se señala "NOTA 1" y "NOTA 2"? ¿Para qué son necesarias estas líneas?
 - Estas líneas cambian el estado de la red neuronal [`net.eval()`] para evaluación y [`net.train()`] para entrenamiento, lo que ajusta el comportamiento de la red según el modo activo.